

CS Capstone: Policy and Process for managing severely compromised or non-performing teams

Executive Summary:

In the real world, “poor performing” consulting companies are rare simply because the open market takes care of it: if your outfit disappoints clients, it just won’t be around for long. In a world where reputation is everything and word spreads quickly, just one lame project execution can mean the end.



In the context of Capstone, this natural and healthy mechanism doesn’t function as clearly: your team is doing just this one project, so doing a lame job has no consequences for your “company’s” future. The reputation of the CS Capstone program is, however, on the line, as well as the integrity of our CS program as whole. If Capstone clients are subjected to teams that are perpetually poor communicators, make little or no progress towards a viable product, and are generally a waste of the client’s time, then that client will certainly never sponsor another Capstone, and will have nothing positive to say about the integrity of NAU’s CS programs. All of the arguments for ensuring strong performance at the individual level also apply to the team level as well: Capstone is designed as a semi-independent consulting experience, and to pass the course, you need to demonstrate that you can function effectively as a team to satisfy a real world client.

Background: Why do we need a disciplinary process? What behaviors warrant action?

Fortunately, having an entire team perform so poorly that an entire project is threatened with failure is rare. Although many teams may have bumps in the road in the early phases (you’re learning, after all), most teams soon establish a strong positive relationship with their clients based on close attention to the client’s input, strong communication, and demonstration of technical competence resulting in consistent project progress.

This policy addresses those cases where a team, as a whole, persistently performs so poorly that they are in danger of complete project failure, i.e., it becomes apparent that the team may substantially fail to complete even the minimal viable product specified by the client. This generally can only happen if multiple (or even all) members of a team are not able to complete their tasks over a period of weeks or months due to some combination of weak effort, poor teaming skills, and lack of adequate technical competencies. In particular, this goes beyond a single team member having performance problems, which is better dealt with using the “Policy for non-performing team members”; in such cases, other team members can generally pick up the slack, shield the client from the internal team problems, and can still substantially complete the project.

In cases where a team has demonstrated dysfunction to the point of impending project failure, the main goal becomes to “cut the losses”, i.e., to avoid wasting more of everyone’s time and protect the integrity of the program. A main emphasis is one preserving the client’s time and good will,

but this also extends to the team mentor's and even that of team members themselves. When a team fails and the "plug is pulled" on a project, this means is that:

- a) The project is terminated. The client is informed (with regrets) that, due to lack of competencies expected of the CS seniors in the team, the project has been cancelled. The client is invited to have priority access to a different team in the next Capstone round.
- b) Having failed to demonstrate central learning outcomes of Capstone, the team members on the project receive an "F" grade for the Capstone course. In cases where a minority of team members did show clear, documented evidence of adequate performance, they may be transferred to other projects/teams at the discretion of Capstone faculty.

Obviously, pulling the plug on a Capstone project is an extremely serious event, with negative consequences all around: team members must repeat Capstone, the client is inevitably disappointed by Capstone and NAU CS, and the CS program must question how it was possible for such a poor team to even arrive at the Capstone level. Thus, this is not a decision taken lightly: there are strong requirements for extensive documentation of poor performance, and teams are clearly warned at multiple occasions of impending disaster.

The goal of the policy and process described here is very much constructive: the emphasis is on making teams aware of their shortcomings in time to address them, and offering mentoring and guidance in correcting the downward trajectory. The absolute best outcome for all parties is a passing score in the Capstone course, and a successful Capstone project.

Corrective Process for Dysfunctional Capstone Teams

Learning to work effectively as a team, including dealing effectively with performance problems and team dynamics, is one of the key learning goals of the Capstone sequence. Even in professional practice, the first aim in any company will always be to help a struggling team identify and address their problems, adjust their internal management practices and personal commitments to the project, and get back on course as a successful team. Only when these mentoring measures fail are more serious disciplinary actions invoked. Accordingly, the process described below begins with warnings and offers to help "debug" team dynamics from the team's mentor, with the goal of getting the team back on a positive course to a successful project completion.

The corrective process consists of the following, increasingly serious steps:

Phase 1: Verbal Warning and Increased Mentoring Intervention. Clearly highlighting team progress/performance issues and offering added mentoring.

When a team's performance/progress slips, it's generally pretty obvious to the whole team: deadlines get missed, obviously poor and unprofessional deliverables get submitted, development progress is slow, and nobody is properly prepared for client meetings. An effective team will react to this immediately, with focused internal team meetings to figure out what's going wrong and addressing it. When it becomes clear that a team is not doing this, i.e., *a consistent trend of poor performance that indicates a project failure trajectory becomes visible*, the first step is for the team mentor to get more involved. Some examples include:

- Clearly and explicitly raise dysfunctional performance in the weekly mentor meetings, verbally highlighting and discussing the trend of poor performance. The idea is to bring the issue out into the open, force discussion of what's going wrong, and identify how the team might repair the situation. This intervention will be explicitly noted on task reports by the team mentor.
- If it appears that initial repair efforts are not effective, the team mentor will send a formal "Memo of Concern" to the entire team by email, CC'ing the CS Faculty Program Organizer. This memo will outline specific evidence and indicators of team dysfunction, e.g., specific deliverables missed, specific failures to meet weekly task deadlines, or other examples of incompetence or inadequate conduct attributable to the team as a whole. The Memo will be explicitly reviewed by the team and the mentor in the next Weekly Team Meeting, to make sure that everyone understands what the issues are... and to actively work towards improvement.
- After one or more "Memos of Concern" with no clear evidence that issues have been resolved, the team mentor will write a "Notice of Team Non-Performance" in which continuing problems are again documented, and the team is asked to prepare a formal "improvement plan" in response. This plan will bullet each performance issue (e.g. late deliverables, poor quality, etc.) outlined in the team mentor's Notice of Non-Performance" and, for each one, will explicitly outline the measures that the team is taking to address the issues. Each subsequent weekly task report prepared by the team will then be extended by adding the issues identified under the "Issues" section of the report; and these issues and how the team is doing on them will be reviewed explicitly as part of each Weekly Team Meeting with the team mentor.

Again, the goal of this process is to raise awareness and, if dysfunction persists, to begin a clear documentation of warnings given and of positive intervention efforts. Most team progress/dysfunction issues will be resolved at this stage, as the team works with their team mentor and other CS faculty to learn from mistakes and get the team and project back on track..

Phase 2.1: Direct Measures. Invoking a professional, targeted disciplinary action to highlight and address continuing performance problems.

At some point, poor productivity/quality will begin to raise the possibility that so much time has been lost/wasted that there is a danger that, even if corrective measures are taken, it's nearly impossible for the team to catch up and deliver a minimally viable product to the sponsor. If a team continues to fail to meet progress and/or quality expectations and project failure is nearly unavoidable, it's time for one last try to remedy the situation before giving up on the project.

The CS Faculty Organizer (in collaboration with the team mentor) writes a final Notice of Impending Termination to the team, with CC to the team mentor and to the CS Department Chair. This memo once again outlines the key issues and the team's repeated failures to improve, and emphasizes that the project is on the brink of falling so far behind as to be

unsalvageable. It states clearly that the project is in imminent danger of being terminated and calls for emergency action by the team

This final Memo of Concern is to be issued at a point where the team has failed in all previous corrective efforts, one can foresee a point in near future where no realistic amount of effort could salvage the project in team...but where this point has not yet been reached. That is, there still remains a chance that, *by an intensive investment of effort substantially above the normal effort expected for Capstone teams*, the team could conceivably make up lost ground and provide at least a minimally respectable deliverable to the client by the end of Capstone. This is a dire last chance situation. A team receiving this memo is formally on notice that there project is headed for termination.

Phase 2.2: Formal Response and Resolution. The team provides explanation and commits to specific resolution.

In addition to documenting grossly inadequate performance, the Notice of Impending Termination formally asks the team to develop a clear “emergency plan” that clearly demonstrates, if executed completely and in timely fashion, could conceivably result in at least a minimally viable product for the client. This is not a memo outlining self-improvement measures for team function; that’s been done in a past step. What is required at this point is a specific plan for catching up lost ground, consisting of clearly measurable steps or milestones. This can be in any form (e.g. a Gantt chart) that clearly shows, for each area/functionality where the team is behind, *specific tasks* that will be accomplished in order to get caught up, *specific persons assigned* to accomplish each task, *an exact date that each task is due on*, and what *specific functionalities* will be demo’ed as evidence that the task is complete.

As a whole, the plan given in the response memo must demonstrate clearly and specifically that the project is still salvageable and the timeline for a sequence of milestones that would achieve this.

Phase 2.3: Faculty Review of Problem and Resolution.

This memo is reviewed for completeness and viability by Capstone staff; if needed, revisions are requested of the team until a concise, measurably verifiable, and complete plan is produced. The final version will then be signed by all team members and becomes a performance contract for the team that, if not met in any aspect, is evidence that the team is incapable of salvaging a passing effort (minimal viable product), and the project should be terminated.

If the team is able to make internal changes, invest the required effort, and get the project back on track, then integrity is restored, the project is allowed to finish as normal with product delivery to the client, and the matter is closed.

Phase 3: Project Termination and failure of non-performing team members.

If any or all of the specific, measurable “catch up” tasks and milestones promised by the team in their response memo are not achieved, then this failed commitment will serve as final evidence that the team has not met the learning goals of Capstone, and the project will be formally terminated. The final decision will be made based on review of the case by the team mentor, the CS Faculty Capstone Organizer, and the Department Chair (or designated representative). If the situation is considered hopeless, the project will be declared as terminated and team members will be informed of their failing grade in Capstone.

Obviously, terminating a project and assigning failing grades -- just as firing from a real job ---- is a very serious matter. Thus, every step of the process must be fully and formally documented. Once any step in the process described here is initiated, all parties should keep archival copies of all emails and memos pertinent to the matter, and should take notes on events as they occur.

It should go without saying that this is an internal team matter, and should be treated with confidentiality by all parties. Specifically, it is not appropriate for any team member to complain about or discuss performance difficulties with persons outside the team.

In sum, pulling the plug on a project is a desperate last measure to preserve program integrity and avoid wasting any more time once a team/project has fallen behind to the point where it is simply impossible to catch up and produce even a minimal viable product during the time remaining before the Capstone Conference event. Ideally, the sanctions outlined in this policy will never be fully executed, as problem teams improve and get back on track. Ultimately, however, pulling the plug on a failing project precisely reflects what happens in professional software engineering contexts and remain an option to protect the integrity and quality of the CS programs at NAU.